Page No.
Date

SQL
Structured Query Language

Data : is a raw fact which describes attributes of an Entity

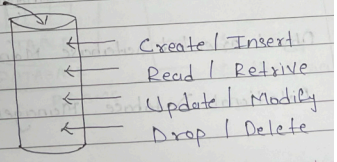
Attributes - Properties
Entity - Object

Entity - is nothing but an object it maybe either living thing or a non-living thing.
- Anything which has physical existen in a real world is known as enti

DataBase - is a place medium or a contains used to store data in a systema & organized manner
- In Database we can perform Sev operations.
- The operations for universely Cal as CRUD operations.

4 tasks / operation

CRUD



Page No.
Date

SQL
Structured Query Language

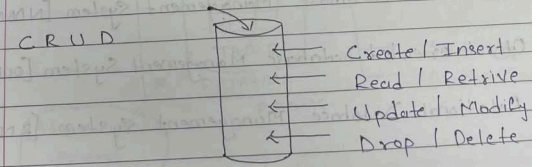
Data :- is a raw fact which describes attributes of an Entity

Attributes - Properties
Entity - Object

Entity - is nothing but an object it may Either living thing or a non-living thing.
- Anything which has physical existe in a real world is known as En

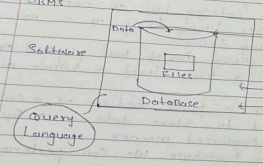
DataBase - is a place medium or a Contai used to store data in a System & organized manner
- In Database we can perform Se operations.
- The operations for Universely C as CRUD operations.

4 tasks / operation



* DBMS (DataBase Management System):- is a Software which is used to maintain and manages the Software Database
- DBMS has two important features
  Security and Authorization
- In DBMS we can Store Data in the form of Files.
- We are using Query language to Communicate with DBMS.

DBMS



Types of DBMS:-

i) Hierarchial Database Management System [HDBMS]

ii) Network Database Management System [NDBMS]

iii) Object oriented Database Management System [OODBMS]

iv) Relational Database Management System [RDBMS]

* Relational Model :- Was designed by Data Scientist E.F.CODD
- Any DBMS which Supports Relational model there will becomes RDBMS.

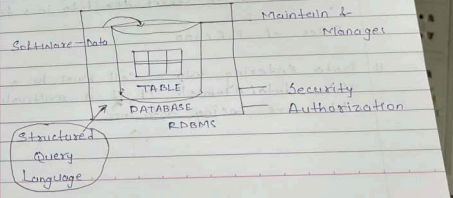
[DBMS] + [Relational Model] ⟶ [RDBMS]

- Any DBMS which Follows the rules of E.F.CODD is known as RDBMS

[DBMS] + [Rules of E.F.CODD] ⟶ [RDBMS]

* RDBMS [Relational DataBase Management System]
- RDBMS is a type of DBMS Software which is used to maintain & manages the database
- In RDBMS we can Store data in the form of tables [Rows & Columns].
- We are using Structured Query Language to Communicate with RDBMS

RDBMS :

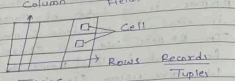


* Difference b/w DBMS & RDBMS

DBMS	RDBMS
- DataBase Management System	- Relational DBMS
- We Can store Data in the form of Files	- We Can store Data in the form of tables
- We are using Query lang to Communicate with DBMS	- We are using Structured Query lang to Communicate with RDBMS.

* Table:- is a logical organization of Rows & Columns and Cells.

Table:



* Cell :- it a intersection between rows & Columns
  - Cell is smallest block or a unit which is used to stores the data in a table.

* Rules of E.F codd

i) Data Entering into a Cell must be a Single Value data (Atomic). If it is multivalued We are facing data loss.

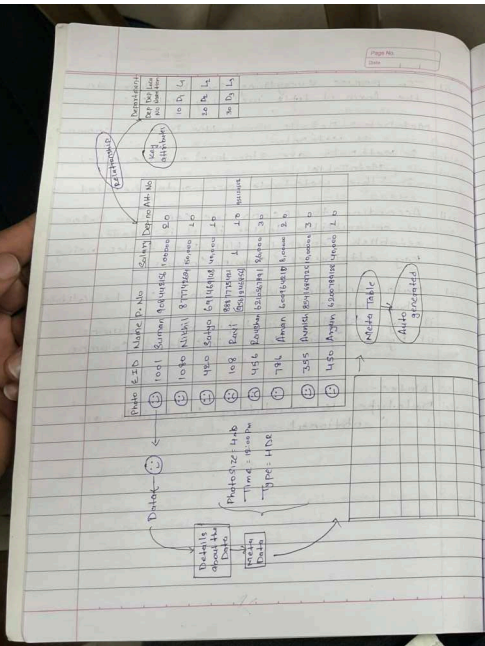
ii) In RDBMS everything Can be stored in the form of table including MetaData.

MetaData :- Details about the Data is known as MetaData.
:- MetaData are stored in a table called MetaTable
:- This Metatable are autogenerated

iii) According to EF Codd We Can store Datas in a multiple tables. if it is needed establish Connection b/w the tables using key attributes (Common Column).

iv) Data entering into a table Can be Validated by two-steps
i) By assigning datatypes to a Columns
ii) By assigning Constraints to a Columns.

Note :-
Datatype is mandatory Constraints are optional.



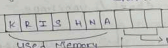
* Datatypes :- It is used to describe 'which kind or type of data should be entering into a particular column or a memory allocation.

-Types of Datatypes:-

(i) Char (ii) VARCHAR/VARCHAR2 (iii) DATE
(iv) NUMBER (v) LARGE OBJECT
→ Character Large object
→ Binary Large object

(i) Char :-
- 'A-Z', 'a-z', '0-9', Special character, Alphanumeric (@, !, $, #, -)
- Characters are must Enclosed with the Single quotations (' ').
- Syntax: Char(size)
- The max. size of char datatype is 2000 Characters
- It follows fixed length of memory allocations

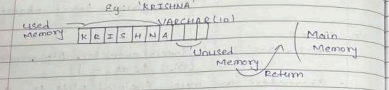
E.g: 'KRISHNA' Char(10)



- In Char datatype unused memory will be wastage of memory.
- Sql is not a Case Sensitive language But Characters datas are Case Sensitive.

(iii) VARCHAR:
- 'A-Z', 'a-z', '0-9', Special Char, 'Alphanumeric' (@, !, $, #, _)
- Characters are must Enclosed with the Single quotation (' ').
- Syntax
  Varchar(Size)

The max Size of varchar data type is 4000.
- It follows Variable length of memory allocation
  Eg: 'KRISHNA'



VARCHAR2:- It is updated version of VARCHAR
- Syntax
  Varchar2(Size)

In VARCHAR2 we can pass 4000 characters as a max size

* Difference b/w Char & VARCHAR2

Char	VARCHAR2
- Char(Size)	VARCHAR(Size)
	VARCHAR2(Size)
- 2000 ch max Size	4000 ch max Size
- Fixed length	Variable length

(iv) DATE:- It is used to store only the Dates
Syntax: Date
- We can store Dates in a oracle specified format
Oracle Format
- 'DD-MON-YYYY' - 'DD-MON-YY'
'01-JAN-25' '31-DEC-2025'

(v) NUMBER:- It is used to store only the numeric Value
Syntax: Number (Precision, [Scale]).
- In Number Datatype we can use 2 args
  i) Precision
  ii) Scale
- Precision is mandatory & Scale is optional.

Precision:- It is used to allocate no of digits required to store an int values.
- The Range of precision is 1 to 38 Digits

Scale:- It is used to store decimal Values within the given precision
- The range of Scales is -84 to 127.
- The default Value of Scale is 0.

Eg-
Number(6): ± 999999
Number(5,3): ± 99.999
Number(3,3): ± .999
Number(3,5): ± .00999
Number(9,10): ± .0099999999

Page No.
Date

iv) Large objects:-

↳ Char Large object:- It is used to store huge amount of characters upto 4gb of size.

Syntax: CLOB

↳ Binary Large object:- It is used to store the Binary values of photo, video, audio, document etc upto 4gb of size.

Syntax: BLOB

* Constraints:- are the rules or a condition which are assigned to the column for validation

Types of Constraints:-

1) Unique
2) Not Null
3) Check
4) Primary Key
5) Foreign Key

1) Unique:- It is a Constraints it cannot accept duplicate or repeated values.

2) Not NULL:- It is a Constraints it cannot accept any Null values.

Page No.
Date

Null:- It is nothing but empty cell.
- It cannot occupies any memory space
- 0 and space cannot be a Null
- If We perform any Arithmetic operation with Null it returns Null value
  e.g- 100 + Null → Null

3) Check:- It is a Constraints which gives extra validation with the help of conditions. If the conditions get satisfied the value or data get enters into a column or else it rejects the data.
  e.g- Check(Salary >= 0)
  5000 > 0
  - Check(Length(Phone_no) = 10)
  2 - 9808480121
  10 = 10

Note
Length() : It is used to count no. of digits or characters.

4) Primary Key:- It is used to identify the unique records from the table.

Characteristics of primary key:-
→ We have only one primary key in a table
→ Primary key cannot accept duplicate or repeated values.
→ Primary key cannot accept null values.
→ Primary key is a combination of Unique & NOT NULL Constraints

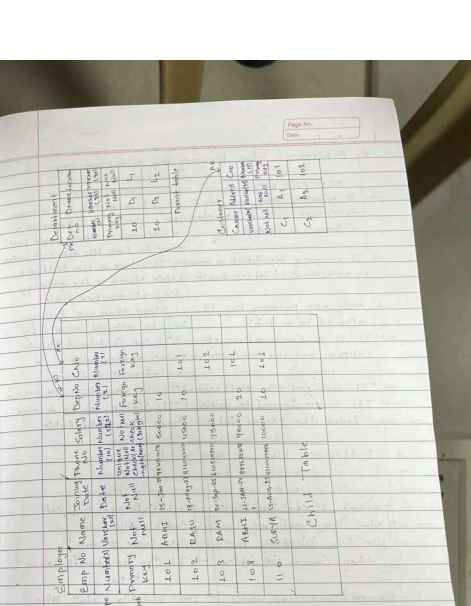
- Primary key is not mandatory But it is highly recommended to used the table.

2) Foreign Key:- It is used to Establish Connection blw the tables.

Characteristics of foreign key:-

- We have n numbers of foreign keys in a table.
- Foreign key will accept duplicate or repeated values
- Foreign key can accept Null values
- Foreign key is not a combination of Unique and Not Null Constraints
- Foreign attributes to become a foreign key it must be primary key in its own table.
- foreign key presenting child table But it actually belongs to parent table.
- Foreign key is also known as referential Integrity Constraints.

* Primary Key	Foreign Key
- It Contains unique Value	- It Contains duplicate Value
- It Cannot Contain null Value	- It Contain null value
- The table has only 1 primary key	- There Can be n no. of foreign key
- used to identify the unique records from the table	- used to establish Connection blw the table.



* Difference b/w Primary key & Foreign key

PRIMARY KEY	FOREIGN KEY
-> A primary key is used to ensure data in the specific Column is unique	-> A foreign key is a column or a group of Columns in a relational database table that provide link b/w data in 2 tables
-> It uniquely identifies a record in relational database table	-> It refers to the field in a table which is the primary key of another table
-> only one primary key is allowed in a table	-> More than one foreign key is allowed in a table
-> It is a Combination of UNIQUE and NotNull Constraints	-> It can Contain duplicate Values and a table in a relational database
-> It does not allow NULL value	-> It can also Contain NULL Value
-> Its Value Cannot be deleted from the parent table	-> Its Value Can be deleted from the child table.

* Difference b/w unique and primary

Unique	Primary
- Cannot accept Duplicate	- Cannot accept Duplicate
- No Combination	- Combination of Unique + NULL
- We Can identify But they Cannot be duplicate	- It Can be identify only 1 item
-> Does not Allows Null Value	- Does not Allow Null Value
-> Can have multiple unique Constraint in a table	- only 1 primary key per table
-> Ensure Values in a Column are unique.	- used to identify the unique records from table.

* overview of Sql statement:
-> Data Definition Language (DDL)
-> Data Manipulation Language (DML)
-> Transaction Control Language (TCL)
-> Data Control Language (DCL)
-> Data Query Language (DQL)

-> Data Query Language (DQL)
"It is used to retrieve or fetch Data from the DataBase

DQL Statements:-
- Select - Display
- Projection - only Column
- Selection - Both Row & Column
- Joins - Data From multiple table

Select:- "It is used to retrive data from the table and display it.

Projection:- It is used to retrive Data from the table by Selecting only the Columns.
- In projection if the Columns are get selected by default the Values present in that Column also get Selected.

Syntax:-

```
SELECT */[DISTINCT] Column_name/Expression[ALIAS]
    FROM Table_Name;
```

order of Execution:
1 -> FROM
2 -> SELECT

Selection:- It is used to retrive datas from the table by selecting both row as well as Columns.

Joins:- It is used to retrive datas from the multiple table simultaneously.

* Write a query to display all the student names present in the student table.

```
Select Sname
From student;
```

student

Sname = SID	Sname = Sname	Branch = Branch	Percentage = Percentage
SID	Sname	Branch	Percentage
1	A	MECH	90
2	B	CIVIL	85
3	C	AERO	67
4	D	ECE	69
5	E	CSE	50
6	F	IT	35

Under execution

Sname
A
B
C
D
E
F

Result table (or) output table

Database

output of From clause

output of Select clause

order of Execution:-

1 -> FROM

2 -> SELECT

NOTES:-

1. In projection, From clause will execute first.
2. In From clause we can pass table name as an arguments.
3. The job of a From clause is goes to the database & makes the table under execution.
4. After the execution of From clause, Select clause will execute.
5. In a Select clause, we can pass 3 arguments
   1) * Astrick
   2) Column name
   3) Expressions
6. A job of a Select clause is goes to an under execution table ~~and~~ search for the column name we have mention.
7. The Select clause is used to fetch the datas from the table at display it.
8. Therefore, the Select clause is responsible for Result table (or) output table.

Commands:-

1) To Set page allignments
-> Set lines 100 pages 100;

2) To clear Screen
-> Cl scr (or) Clear Screen

3) To describe the table structure
-> desc table_name;
   desc student;

* Write a query to display all the details from the database.

```
Select *
From tab;
```

* Write a query to display student ID and percentage branch for all the students

```
Select * SID, branch, Per
From student;
```

* Write a query to display student ID and percentage, branch for all the students

```
Select * SID, branch, Per
From student
```

* Write a query to display student ID, student name, branch & percentage for all the student

```
Select * Sid, Sname, branch, Per
From student;
```

Astrick * :-
It is used to display all the details from table
Along with astrick we can't use any other Column name or expressions

* Write a query to display all the details present in the student table

```
Select *
From student;
```

* Write a query to display all the details from the employee table

```
Select *
From emp;
```

* Write a query to display all the details from the department table

```
Select *
From dept;
```

- Column names:

Employee table (Emp)	Department table (Dept)
EMP NO	DEPT NO
E NAME	DNAME
JOB	LOC
MGR	
HIREDATE	
SAL	
COMM	
DEPTNO	

1) Write a query to display all the details from the emp

```
Select *
From emp;
```

2) Write a Names of all the employees in query
```
Select Ename;
From emp;
```

3) WAQTD Names & Salary given to all the employee
```
Select ename, Comm,Sal
From emp;
```

4) WAQTD Name & Commition given to all the employees
```
Select ename, Comm
From emp;
```

5) WAQTD employee No & Department number of all the employees in emp table
```
Select empno, deptno
From emp;
```

6) WAQTD Ename and hiredate of all the employees
```
Select ename, hiredate
From emp;
```

7) WAQTD name and designation of all the employees
```
Select ename, job
From emp;
```

8) WAQTD name, job & Salary given all the employees
```
Select name, job, sal
From emp;
```

9) WAQTD Dnames present in different table
```
Select dname
From dept;
```

10) WAQTD Dnames & location in different table
```
Select dname, loc
From dept;
```

* Distinct Clause:- It is used to remove duplicate or repeated values from the result table.
- We can pass distinct as a first argument in the select clause. (We cannot use distinct as a second argument in the select clause)
- In distinct We can use multiple column name it will remove duplicate or repeated values from the combination of the columns.

* WAQ Different student name present in the Student table
```
Select Distinct Sname
From Student;
```

Student

Sid	Sname	Branch	Per
1	A	AERO	70
2	B	CIVIL	80
3	C	MECH	75
4	D	ECE	90
5	A	EEE	70
6	E	CSE	65
7	B	CIVIL	80
8	F	Bcom	99

Sname
A
B
C
D
A
E
B
F

output of Select clause

Sname
A
B
C
D
C
D
E
F

4) WAQ different Sname Branch present in student table.
```
Select Distinct Sname, Branch
From Student;
```

Sid	Sname	Branch	Per
1	A	AERO	70
2	B	CIVIL	80
3	C	MECH	75
4	D	ECE	90
5	A	EEE	70
6	E	CSE	65
7	B	CIVIL	80
8	F	BCOM	92

Sname	Branch
A	AERO
B	CIVIL
C	MECH
D	ECE
A	EEE
E	CSE
B	CIVIL
F	BCOM

Sname	Branch
A	AERO
B	CIVIL
C	MECH
D	ECE
A	EEE
E	CSE
F	BCOM

* WAP different student name Branch and percentage from student table

Sid	Sname	Branch	Per
1	A	AERO	70
2	B	CIVIL	80
3	C	MECH	75
4	D	ECE	90
5	A	EEE	70
6	E	CSE	65
7	B	CIVIL	80
8	F	BCOM	92

Sname	Branch	Per
A	AERO	70
B	CIVIL	80
C	MECH	75
D	ECE	90
A	EEE	70
E	CSE	65
B	CIVIL	80
F	BCOM	90

Sname	Branch	Per
A	AERO	70
B	CIVIL	80
C	MECH	75
D	ECE	90
A	EEE	70
E	CSE	65
F	BCOM	90

* WAQT Different, Branch, Per for all student
  Select Distinct Sid, Branch, Per
  From student;

-> Select Sname, distinct Per
  From student; ——— Error

Note:- We cannot use Distinct as a second argument in the Select clause

* WAQTD Different Jobs present in Employee table
  Select Distinct Job
  From student;

* WAQTD Different Jobs & Department no present in employee table
  Select Distinct Job, Deptno
  From student;

* WAQTD Job, Salary, and Hiredate present in Emp table
  Select Distinct Job, Sal, Hiredate
  From Emp;

* Expression

-> Any statement which gives result is known as Expression

Ex

5 + 5 = 10 —> Result

Expression

Expression is always the Combination of operands and operator

Sal * 12
operand operators

operands :- may be a Column name or Value

operators :- are the predefined Symbols used to perform specific task

* WAQTD annual Salary for all the Employees
  Select Sal*12
  From emp;

* WAQTD employe name & Hait term Salary for all employees.

```
Select ename, Sal*6
From emp;
```

* WAQTD Salary & Salary with hike 300 for all emp.

```
Select Sal, Sal+300
From emp;
```

* WAQTD name, Job & Salary with deduction of 500 for all employee.

```
Select name, Job, Sal-500
From emp;
```

* WAQTD Salary with hike of 10% for all employee.

```
Select Sal+Sal*0.1
From emp;
```

Note:- % with hike (or) Deduction

$$Sal + Sal \times \frac{K}{100}$$

$$Sal - Sal \times \frac{K}{100}$$

* WAQTD annual Commision & annual Salary deduction of 35% for all emp.

```
Select Comm*12, Sal*12-(Sal*12*35/100)
From emp;
```

↳ $Sal \times 12 + Sal \times 12 \times \frac{K}{100}$

↳ $Sal \times 12 - Sal \times 12 \times \frac{K}{100}$

* Details of employe, Annual Salary for all employe

```
Select *
```

Note:- If we want to Display all the details along with column name or expression we can use tablename.*

* [ALIAS] :- it a alternative name given to the Columnname or expression in the result table

-> We can assign Alias with or without using keywords

-> As keyword

-> Alias will be a single string

-> If it is a multiple string we can use Double quotation (" ") or Underscore (_).

```
Select Sal*12 As Annual_Salary
From emp;
```

Eg:

```
Sal*12 Annual_Salary
Sal*12 "Annual Salary"
Sal*12 As "Annual Salry"
Sal*12 As "Annual_Salary"
```

* WAQTD Annual Salary & Annual Commision with deduction of 100 Rs for All the employees.

```
Select Sal*12 As "Anual sal", Comm*12-100
As "Annual Comm Deduction"
From emp;
```

* WAQTD employee name & Salary with hike of 1000 for all the employees.
Select ename, Sal+1000 As hike
From emp;

* WAQTD details of an employee along with annual Salary & annual Commission for all the employee.
Select emp.*, Sal*12 As "Annual Sal", Comm*12 As "Annual Comm";
From emp;

ALIAS Query:-

1. WAQTD Name of the employee along with their Annual Salary.
-> Select name, Sal*12 As "Annual Sal"
From emp;

2. WAQTD EName and Job for all the employee with their half term Salary.
Select ename, Sal*6 As "Half term Salary"
From emp;

3. WAQTD All the details of the empl along with an Annual Bonus of 5000.
Select emp.*, Sal*12 +5000 as "Annual Salary"
From emp;

4. WAQTD Name Salary and Salary with hike of 10%
Select ename, Sal, Sal*10/100 as "hike"
from emp;

5. WAQTD Name and Salary with Deduction of 25%
Select ename, Sal, Sal-Sal*25/100 As "Sal deduction"
from emp;

6. WAQTD Name and Salary with monthly hike of 50
Select ename, Sal+50 as "Hike"
from emp;

7. Name & Annual Salary with deduction of 10
Select ename, Sal*12, Sal*12 - (Sal*12*10/100) As "Annual Salary deduction"
From emp

8. Total Salary given to each employee
Select Sal+comm as "Total Salary"
from emp;

9. Details of the emp along with annual Salary
Select emp.*, Sal*12 As "Annual Salary"
from emp;

10. Name & Designation with 100 pentel in Salary
Select ename, job, Sal+100 as "Pentenity"
from emp;

* Selection
-> It is used to retrive data from table by selecting both rows & columns

Syntax:-
Select */ [DISTINCT] Column name | Expression [ALIAS]
From table name;
[Where filter conditions]

Where Clause:- it used to filter the records
- Where Clause executes after the from clause
- Where Clause executes Row by Row
- In Where Clause We Can use filter Condition as an argument.
- Filter Condition
  Column name / Expression operator Value/Data
- It Returns Boolean Values (T|F)
- In Where Clause We Can Write multiple Condition With the help of logical operators.

* WAQTD Name & DeptNo of employees if the employees are Working in DeptNo 10.

```
Select ename, Deptno
From emp
Where Deptno=10;
        Filter Condition
```

output of From Clause

Empno	Ename	Sal	Deptno
1	SCOTT	1500	30
2	JAMES	950	10
3	ALLEN	3500	20
4	MILLER	1950	10
5	WARD	2750	30
6	SMITH	1250	20
7	KING	4000	10

Under Execution

Filter Condition Deptno=10

1	SCOTT	1500	30	
✓ 2	JAMES	950	10	
3	ALLEN	3500	20	
✓ 4	MILLER	1950	10	
5	WARD	2750	30	
6	SMITH	1250	20	
✓ 7	KING	4000	10	

output of Where Clause

Empno	Ename	SAL	Deptno
2	JAMES	950	10
4	MILLER	1950	10
7	KING	4000	10

output of Select Clause

Ename	Deptno
JAMES	10
MILLER	10
KING	10

* WAQTD Name & Salary of the employees if employees are Earning Salary less than 2000.

```
Select ename, Sal
From emp
Where Sal<2000;
```

* WAQTD Details of an employees where Working as a manager.

```
Select *
From emp
Where Job='MANAGER';
```

* WAQTD Name, Job & HireDate of employees if Employees Are Hired After 31st Dec 82.

```
Select ename, Job, HireDate
From emp
Where HireDate > '31-DEC-82'
```

* WAQTD Details of employee along with annual Sal if annual Sal is less than 25000.

```
Select emp.*, Sal*12 As "Annual Salary"
From emp
Where Sal*12 < 25000;
```

Note:- We Cannot use ALIAS Name in Where Class Because Where Clause Executes Before Select Clause.

* WAQTO Job & HIREDATE of Emp if the Employees are Hired Before the year 82

Select JOB, HIREDATE
From EMP
Where HIREDATE < '01-JAN-82';

NOTES:-
DATE: 'DD-MON-YY' --> Before the year 82('DD-MON-82')
Hired Before the year 82
Hired Before the 1st Date of year 82
Hired Before the 01-JAN-82
- < '01-JAN-82' Less than

Hired After the year 82
Hired After the Last Date of the year 82
Hired After the 31-DEC-82
> '31-DEC-82 Greater than

* WAQTO Details of employee whose name is MILLER

Select emp*
From emp
Where ename = 'MILLER';

* WAQTO Details of employee & Salary with hike of 15% if the emp are earning Salary Rs 3000 Rs

Select emp.*, SALT SALVIS As "SAL HIKE"
100
From emp
Where SAL = 3000;

Assignment

1. WAQTO Annual Sal of Emp Whos Name is SMITH

Select * Sal * 12 As annual = Sal
From emp
Where ename = 'SMITH';

2. WAQTO NAME of Emp Working As CLERK

Select Ename, ~~Job~~
From emp
Where Job = 'CLERK';

3. WAQTO SALARY of Emp Working As SALESMAN

Select SAL, ~~Job~~
From emp
Where Job = 'SALESMAN';

4. WAQTO Details of Emp Who earns more than 2000

Select *
From emp
Where Sal > 2000;

5. WAQTO Details of Emp Who was Hired After 01-JAN-81

Select *
from emp
where hiredate > '01-Jan-81';

6. WAQTO Details of Emp Whos name is JONES

Select *
from emp
where ename = 'JONES';

7. WAQTD NAME and SAL Along with his Annual Salary if the annual Salary is more than 12000

```
Select ename,sal,sal*12 as Annual_Sal
from emp
where sal*12 > 12000;
```

8. WAQTD EMPNO of the Employees who are Working in DEPT 30.

```
Select empno
from emp
where Deptno=30;
```

9. WAQTD ENAME and HIREDATE if they are Hired before 1981

```
Select ename,hiredate
from emp
where hiredate < '01-JAN-81';
```

10. WAQTD Details of emp Working As MANAGER

```
Select *
from emp
where Job='MANAGER';
```

11. WAQTD NAME and SALARY. If Emp Earns Commission of Rs 1400

```
Select ename,sal
from emp
where comm=1400;
```

12. WAQTD DETAILS of Emp Having Commision more than Salary

```
Select *
from Emp
where comm > sal;
```

13. WAQTD EMPNO of Employees HIRED Before the year 87

```
Select empno
from emp
where hiredate < '01-Jan-87';
```

14. WAQTD Details of Employees Working as an Analyst

```
Select *
from emp
where Job='ANALYST';
```

15. WAQTD Details of Emp Earning more than 2000/Month

```
Select *
From emp
where sal>2000;
```

x. Operators In Sql

1. Arithmetic operators :- (+, -, *, /)
2. Concatenation operator :- (||)
3. Comparison operator :- (=, !=, or <>)
4. Relational operator :- (>, <, >=, <=)
5. Logical operator :- (AND, OR, NOT)
6. Special operator :- (1. In
   2. NOT IN
   3. BETWEEN
   4. NOT BETWEEN
   5. IS
   6. IS NOT
   7. LIKE
   8. NOT LIKE).
7. Subquery operators :- (1. ALL
   2. ANY
   3. EXISTS
   4. NOT EXISTS)

* Concatenation operator :- It is used to join the given string.
- The symbol for concatenation operator is '||'

* Select 'MR ' || ENAME
From EMP;
'MR ' || ENA
MR SMITH
MR ALLEN

* HII SMITH UR SALARY IS 800
Select 'HII' || ENAME || 'UR SALARY IS' || SAL
From EMP;

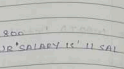
* -- HI SMITH UR OF NAME IS SHEELA --
Select 'HII' || ENAME || 'UR OF NAME IS SHEELA'
FROM EMP
WHERE ENAME = 'SMITH';

* Logical operator :- It is used to write multiple conditions in a where clause.
(AND, OR, NOT)

- AND operator :- It is a Binary Multiplication operator
- AND operator return true if all the conditions get satisfied

A	B	A AND B
T	F	F
F	T	F
F	F	F
T	T	T

- OR operator :- It is a Binary Addition operator
- OR operator returns true if anyone of conditions get satisfied



A	B	A+B
T	F	T
F	T	T
T	T	T
F	F	F

- NOT operator :- It is a Unary operator and it is also known as inverse operator.
- NOT operator is similar to Unequal operator
- It is used to inverse the result / oppose the result

A	$\bar{A}$
T	F
F	T

Deptno != 10
NOT Deptno = 10

* WANTD Details of emp if the emp are working as manager in Deptno 10
Select *
from emp
where Job = 'MANAGER' AND DEPTNO = 10;

* WNATD Name, Job & Sal of emp if emp are working as a Salesman & Earning Salary greater than 1000 & Dept no 30
Select ename, Job, Sal
From emp
where Job = 'Salesman' AND Sal > 1000 AND DeptNo >= 30;

* WAQTD Name & Salary of emp if emp are earning Salary greater than 1500 and less than 3000

```
Select ename, Sal
from emp
where Sal > 1500 AND Sal < 3000;
```

* WAQTD Name & Job of emp if the emp are working as a Salesman or analyst

```
Select ename, Job
from emp
where Job = 'Salesman' or Job = 'analyst';
```

* WAQTD Details of emp working as a President or earning Sal 5000

```
Select *
from emp
where Job = 'President' or Sal = 5000;
```

* WAQTD Details of emp except who working Deptno 20

```
Select *
from emp
where Deptno != 20;
    or
where Not Deptno = 20;
```

* WAQTD Details of emp except who are earning Sal > 1500

```
Select *
from emp
where NOT Sal > 1500;
```

* WAQTD Details of emp who are working as a manager in Deptno 10 or 20

```
Select *
from emp
where Job = 'manager' AND (Deptno = 10 or Deptno = 20);
```

* WAQTD Details of emp along with annual Sal if emp are working as a Salesman or clerk in Deptno 30

```
Select *, Sal*12
from emp
where (Job = 'Salesman' or Job = 'clerk') AND Deptno = 30;
```

* WAQTD Name, Job, Deptno of an emp if emp are working as analyst or Salesman in Deptno 20 or 30

```
Select ename, Job, Deptno
from emp
where (Job = 'analyst' or Job = 'Salesman') AND (Deptno = 20 or Deptno = 30);
```

* WAQTD Details of an emp except who working Deptno 10 or 20

```
Select *
from emp
where NOT (Deptno = 10 OR Deptno = 20);
```

* WAQTD Details of emp except who working as manager or Analyst

```
Select *
from emp
where Not (Job = 'MANAGER' or Job = 'Analyst');
```

* WAQTD Details of emp working as clerk and earning < 1500

```
Select *
from emp
where Job = 'clerk' AND earning < 1500;
```

2. WAQTD Name, Hiredate of emp working Manager In Dept 30

```
Select ename, hiredate
from emp
where Job='manager' AND Deptno=30;
```

3. WAQTD Details, Annual Sal if working in Deptno 30 As Salesman and annual Sal = 14000

```
Select emp*, Sal*12
from emp
where Job='Salesman' AND Deptno=30 AND Sal>14000;
```

4. WAQTD All the Details of emp working Dept 30 OR as Analyst

```
Select emp.*
from emp
where Deptno=30 or Job='Analyst';
```

5. WAQTD Names of emp whose Sal is < 1100 AND Their Job is Clerk

```
Select ename
from emp
where Sal<1100 AND Job='clerk';
```

6. WAQTD Name, Sal, Annual Sal and Deptno if Deptno is 20 earning more than 1100 and Annual Sal exceeds 12000

```
Select ename, Sal, Sal*12, Deptno
from emp
where Deptno=20 AND Sal>1100 AND Sal*12>12000;
```

7. WAQTD Empno & Names of emp working as Manager in Dept 20

```
Select ename, empno
from emp
where Job='manager' AND Deptno=20
```

8. WAQTD Details of emp working in dept 30 or 20

```
Select *
from emp
where Deptno=30 or Deptno=20;
```

9. WAQTD Details of emp Working as Analyst in Deptno 10

```
Select *
or from emp
where Job='Analyst' And Deptno=10
```

10. WAQTD Details of emp working as President with Sal 5000

```
Select *
from emp
where Job='President' AND Sal='5000';
```

Assignment

* WAQTD Names and Deptno Jobs of Empl working as Clerk In Dept 10 or 20

```
-> Select ename, Deptno, Job
   From emp
   Where Job='clerk' AND (Deptno=10 or Deptno=20);
```

* WAQTD Details of employees working As Clerk In Dept 10 or manager

```
Select *
From emp
Where (Job='clerk' or Job='Manager') AND Deptno=10;
```

* WAQTD Names of employees working in Dept 10, 20, 30, 40

```
Select ename
From emp
Where Job Deptno=10 AND Deptno=20 OR Deptno=30 or Deptno=40;
```

* WAQTO Details of employees with empno 7902, 7839

```
Select *
From emp
Where empno=7902 OR empno=7839;
```

* WAQTO Details of employees working as manager or Salesman or clerk

```
Select *
From emp
Where Job='Manager' or Job='Salesman' or Job='clerk';
```

* WAQTO Names of employees Hired After 81 AND before 87

```
Select ename
From emp
Where hiredate>'31-DEC-81' AND hiredate<'1-JAN-87';
```

* WAQTO Details of employees Earning more than 1950 but less than 3000

```
Select emp.*
From emp
Where Sal>1950 AND Sal<3000;
```

* WAQTO Names of employees Hired after 81 Into Dept no 10 or 30

```
Select ename
From emp
Where hiredate>'31-DEC-81' AND (Deptno=10 or Deptno=30);
```

* WAQTO Names of employees Along With Annual Salary for emp working as manager or Clerk into Dept no 10 & 30.

```
Select ename, Sal*12
From emp
Where (Job='manager' or Job='clerk') AND (Deptno=10 or
                                          Deptno=20);
```

* WAQTO All the Details along with Annual sal if Sal is between 1000 AND 4000 Annual Salary more than 15000

```
Select emp.*, Sal*12
From emp
Where Sal>1000 AND Sal*12>15000;
```

* WAQTO Details of all the employees except the emps working in Dept 10 or 30

```
Select *
From emp
Where Deptno!=10 AND Deptno!=30;
```

* WAQTO Details of all emps along with Annual Sal except the Employees Working As Analyst or Manager

```
Select emp.*, Sal*12
From emp
Where Job!='Analyst' AND Job!='Manager';
```

Special operators:- (1. IN
2. NOT IN
3. BETWEEN
4. NOT BETWEEN
5. Is
6. Is NOT
7. LIKE
8. NOT LIKE)

* In operator:- In Similar to Equals operator.
- In operator Is known as Multivalued operator
- It Can accept multiple values belongs to their Same Column in RHS.

- IN operator returns true if anyone of Condition get satisfied
- Syntax:
  Column-name/expression IN(V1,V2,... Vn)

* WAQTO Name of an employee if employee are working in a Department id 10 or 30.

Ename	Sal	Deptno
ALLEN	1250	30
MILLER	1500	10
FORD	9000	40
WARD	8700	20
KING	1900	10

Select Ename
From Emp
Where Deptno IN(10,20,30);

* WAQTO Details of emp if Emp working as manager or Analyst and earning Sal >1000.
  Select *
  from emp
  Where Job IN('Manager','Analyst') AND Sal >1000;

* WAQTO Job & Deptno of emp if emp are working as a Salesman or Analyst in Deptno 10 or 30
  Select Job, Deptno
  from emp
  Where Job IN('Salesman', 'Analyst') AND Deptno IN(10,30);

* NOT IN: operator is similar to IN operator instead of selecting the values it rejects the value
  Syntax
  Column-name / Expression NOT IN (v1, v2 ... vn);

* WAQTO Details of an employee except who are working in Deptno 10,30
  Select *
  from emp
  Where Deptno NOT IN (10,20);

* WAQTO Details of an employee except who are working as a manager or Analyst in Dept 10 or 30
  Select *
  from emp
  Where Job NOT IN ('manager', 'ANALYST') AND
  ~~and~~ DEPTNO IN (10,20);

* BETWEEN OPERATOR: Whenever we have Range of values we use BETWEEN operator
  (Starting Value & Ending Value)
- Between operators works including the ranges
- We cannot interchange the Ranges in Between operator
  Syntax
  Col name / Exp ... Between Lower rang and higher rang

* Name of emp who Sal > 1500 and Sal < 3000
  Select Ename
  from emp
  Where Sal Between 801 And 2999;

Eg:

1250 Between 1501 And 2999 = F
<1501

1500 Between 1501 And 2999 = F
=1501

3000 Between 1501 And 2999 = F
>2999

1700 Between 1501 And 2999 = T
>1501 <2999

1300 Between 1501 And 2999 = T
>1501

Note:-

Sal > 1500 And Sal < 3000

Range: 1501 And 2999

Sal in the Range of 1500 to 3000

Range: 1500 and 3000

Salary Between 1500 to 3000

Range: 1501 And 2999

* WAQTD Details of employe if employees are hired during the year 81

```
Select *
from emp
where hiredate Between '01-JAN-81' AND '31-DEC-81';
```

* WAQTD Name & Salary of Emp who are Earning Sal in Range of 1250 to 3000

```
Select ename, Sal
from emp
Where Sal Between 1250 and 3000;
```

* WAQTD Name & Hiredate of emp if emp are hired after the year 81 But Before the year 89 But Before the year

```
Select ename, hiredate
From emp
Where hiredate Between '01-Jan-82' and '31-dec-88';
```

- NOT Between:- It is similar to between operator instead of selecting the ranges it reject the range of values.

Syntax

Column name | expression NOT BETWEEN lower-range AND higher-range

* WAQTD Details of employees except who Earning Sal in range of 1250 to 2500

```
Select *
from emp
Where Sal not Between 1250 and 2500;
```

* WAQTD Name & Salary of emp except Earning Sal less than 1000 and greater than 3000

```
Select ename, Sal
from emp
Where Sal not between 1000 and 3000;
```

* WAQTD hiredate of emp except who are hired during yr of 81

```
Select hiredate
from emp
where hiredate not between '01-Jan-81' And '31-Dec-81';
```

- Is : It is used to Compare with only the null values.
Syntax:
Column name / expression Is Null;

* WAQTD name & Comm of emp if emp are not earning any Comm.
Select ename, Comm
from temp
Where Comm is null;

+ WAQTD Details of emp if emp doesn't have any manager no.
Select *
From emp
Where mgr is null;

- Is not: It is used to Compare with not null values
Syntax:
Column name / expression Is NOT NULL;

* WAQTD Details of an emp who earn some Comm
Select *
From emp
Where Comm is not null and Comm != 0;

+ WAQTD Details of if emp are earning some Sal but not Comm.
Select *
From emp
Where Sal is not null and Comm is null;

- Like :- It is used to matching the patterns.
Syntax:
Column name / expression Like 'Pattern to match';

- To matching the patterns in Like operator we use 2 special characters
i) % (Percentage)
ii) _ (Underscore)
i) Percentage(%): It can accept n number of characters any characters But sometimes no characters.
ii) Underscore(_): The one underscore can accept exactly only one character but any type characters.

* WAQTD Names of emp if emp names starts with character 'A'
Select ename
From emp
Where ename Like 'A%';

+ WAQTD Details of emp if emp name ends with character 'R'
Select *
From emp
Where ename Like '%R';

x WAQTD Name of emp where name starts with 'J' and 'S'.
Select ename
From emp
Where ename Like 'J%S' AND 'J%S';

* WAQTD Names of emp if the emp names has ch 'A'

```
Select ename
from emp
where ename like '%A%';
```

* WAQTD Name & hiredate of emps if emps are hired during 81

```
Select ename, hiredate
from emp
where hiredate like '%81';
```

* WAQTD Name & hiredate of emps if emps are hire in month Dec

```
Select ename, hiredate
from emp
where hiredate like '%DEC%';
```

* WAQTD Details of employee, if the employee name has exactly 5 character

```
Select *
from emp
where ename like'_____';
```

* WAQTD emp name, the emp name A as a 3rd character

```
Select ename
from emp
where ename like '%A%';
                 '__A%';
```

* WAQTD name, Sal of emp if the emp E char at last

```
Select ename, Sal
from emp
where ename like '%E_';
```

* WAQTD name of emp, if emp name has second char 'A' and 'r' char is 4

```
Select ename
from emp
where ename like'_A_E%';
```

* NOT LIKE:-

- It is similar to like operator but it will reject the pattern

Syntax: Col.name/expression NOT LIKE 'Pattern-to-match';

* WAQTD name, Sal of emp except 4 digit Sal

```
Select ename, Sal
from emp
where Sal NOT LIKE '____';
```

* WAQTD name of emp except those name end with ER

```
Select ename
from emp
where ename NOT LIKE '%ER';
```

* WAQTD hiredate of an emp exce who are hire during 81

```
Select hiredate
from emp
where hiredate NOT like '_____81';
                        '%81';
```

* WAQTD name of emp except those name has char S

```
Select ename
from emp
where ename NotLike '%S%';
```

1 WAQTD names of emp if the emp starts with Vowels

```
Select ename
from emp
where ename like 'A%' or 'E%' or 'I%' or 'O%' or 'U%';
```

2 WAQTD name of emp if emp name has 2 time 'A' in their name

```
Select ename
from emp
where ename like '%A%A%';
```

3 WAQTD name if the name has Exactly 2A in present in name

```
Select ename
from emp
where ename like '%A%' And ename not like '%A%A%';
```

4 WAQTD Details of emp whose name starts with 'A'

```
Select *
from emp
where ename like 'A%';
```

5 WAQTD Details of emp whose name end with 's'

```
Select *
from emp
where ename like '%s';
```

6 WAQTD Details of emp whose name starts with 'A' and with 's'

```
Select *
from emp
where ename like 'A%%s';
```

7 WAQTD Details of emp whose name start with 'A' & end with A

```
Select *
From emp
where ename like 'A%%A';
```

8 WAQTD Details of emp whose name has Char R in it

```
Select *
From emp
where ename like '%R%';
```

9 2-as 2&4 char in name

```
Select *
From emp
where ename like '_A%'; or '%AA%';
```

10 Details of emp I in 3rd char

```
Select *
From emp
where ename like '__I%' or '%I_%';
```

11 A-1st, M-last But (2nd last)

```
Select *
From emp
where ename like 'A%M%' or 'A%M_';
```

12 Exactly 4 Char

```
Select *
From emp
where ename like '____';
```

10. ename, hiredate of emp hire During 82

```
Select ename, hiredate
from emp
Where hiredate like '%82' or '%-82';
```

11. ename, hiredate of emp hire in month of Feb

```
Select ename, hiredate
from emp
Where hiredate like '%Feb%';
```

12. ename, Sal of emp Who are Earning Sal of 3 Digit

```
Select ename, Sal
from emp
Where Sal like '___';
```

13. ename, hiredate and Sal of emp Who Earn Sal in 3 digit & Whole hired in the month in Dec

```
Select ename, hiredate, Sal
from emp
Where Sal like '___' AND hiredate like '%Dec%';
```

14. ename, Sal of emp Who Sal 4 digit & name start with J

```
Select ename, Sal
From emp
Where Sal like '____' AND ename 'J%';
```

15. Name has 5 char 4th char is E

```
Select ename
From emp
Where ename like '___E_';
```

Assignment:

1. WAQTD Details of emp Working in Dept 10 or 20

```
Select *
From emp
Where Deptno In (10,20);
```

2. WAQTD name of emp hire during 82

```
Select ename
from emp
Where hiredate between '01-JAN-82' AND '31-DEC-82';
```

3. WAQTD Name & Salary give to employee earning Comm

```
Select ename, sal
from emp
Where Comm is NOT NULL;
```

4. WAQTD Details of emp Working as Clerk in Dept 10 or 30 Having Character 'A' in their name

```
Select *
From emp
Where Job = 'clerk' and Deptno In (10,30) and ename like '%A%';
```

5. WAQTD names of emp having char's' As their last cha--

```
Select ename
from emp
Where ename like '%s';
```

6. name, hiredate of emp hired after 1982 But Before 1987

```
Select ename, hiredate
from emp
Where hiredate between '01-Jan-83' and '31-Dec-86';
```

7. WAQTD Details of employee Working as analyst and earning in high sal

```
Select *
From emp
Where Job='Analyst' and Sal>1500;
```

8. WAQTD Names of the employees Hired In The First Month

```
Select ename
From emp
Where hiredate like '%JAN%';
```

9. WAQTD Details of the employees Working As Salesman And Doesn't Earn any Comission

```
Select *
From emp
Where Job='Salesman' And Comm is Null;
```

10. WAQTD Name and Salary Given to the Emp Hired during 1999 In the Dept no 20 or 40

```
Select ename, Sal
From emp
Where hiredate between '01-Jan-1999' and '31-Dec-1999'
Deptno between In (10,40);
```

11. WAQTD Details of Emp having 'L' Presents In The Names and Works as Manager

```
Select *
From emp
Where ename Like '%L%' and Job='Manager';
```

12. WAQTD Names of emp who earn Commission But Not Salary

```
Select ename
From emp
Where Comm is not Null AND Salary is Null;
```

13. WAQTD Names of the emp if Name starts With 'A' or 'J' or 'S'

```
Select ename
from emp
Where ename Like '%A' AND 'NOT' AND '%S';
```

14. List All the Emp Names except for the emp who's Name has 'A' At the 3rd character

```
Select ename
from emp
Where Not Like '__A%';
```

15. List the Details of the emp Working As Managers' and Hired After 1994 and has A Name which Ends With 'S'

```
Select *
from emp
Where Job='Manager' AND Hiredate='01-JAN-1995'
AND ename Like '%S';
```

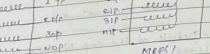
* Functions is the list of Code or block of instruction used to perform specific task

Functions

- In Built Function
- User Define fun

Single Row Function SRF()

Multi-Row → Group function
Function → Aggregate fun
MRF



- SRF() :- It Executes Row by Row
- It We pass one input in a Single Row function in process execute & generates one output.
- It We pass n no. of inputs in Single Row Function it generates n no. of outputs.

Emp

Ename	SAL
ALLEN	1800
MILLER	8500
FORD	1950
TURNER	4700
SCOTT	3850

Length() :- To count the no. of characters

Select Ename, Length(Ename) from Emp;

output

Ename	Length(Ename)
ALLEN	5
MILLER	6
FORD	4
TURNER	6
SCOTT	5

- MRF() (Group Function() / Aggregate Fun())
- Multi-Row Fun Aggregates all the records at a Single shot and Execute and then generates a single output.
- It We pass one input it generates one output
- It We pass n no. of inputs it generates Exactly only one output.

List of Multi-Row-Function

- Max()
- Min()
- Sum()
- Avg()
- Count()

* IN ADDTN max Sal present in emp table

Select Max(Sal)
From emp;

1800
8500
1950
4700
3850

output
Max(Sal)
4700

- Rules of Multi Row Function
- In Multi Row fun we can use single args (implicit)
- The arguments may be Col-name or Expression
- Along with MRF() we cannot use arg Col-name or Expressions
- MRF() ignores the null values
- We cannot use multiRow Fun() in Where clause
- Count is the only multiRow Fun() which accepts Astrick(*) as a argument

* WAQD Max & Min Sal given to all the employees
  Select Max(Sal), min(Sal)
  from emp;

* WAQD No of emps Working as Salesmen
  Select * Count(*)
  from emp
  Where Job = 'Salesman';

* WAQD Avg Sal given to the emps who are hired before the year 1st and Earning Some Comm
  Select Avg(Sal)
  from emp
  Where HireDate Between '01-JAN-82' and '01-DEC-82'
  and Comm is not NULL;

* WAQD No of Emp & Total Sal given to emp if emps are Working as a Salesman or Analyst In dept no 20 or 30
  Select Count(*), Sum(Sal)
  from emp
  Where Job In ('Salesman', 'Analyst') And DeptNo In (20, 30);

* WAQD No of Distinct Job present in emp table
  Select Count(Distinct Job)
  From emp;

* WAQD Total Salary Avg Salary given to the Emp if emps are Earning Some Salary and their name has Exactly 5 chars
  Select Count(Sal), Avg(Sal)
  from emp
  Where Sal is Not Null AND Ename Like '_____';

* Escape Character(*) :- It is used to remove the special behaviour given to a special character ('%, _') and to treat it as a normal character

Syntax:
Col-name / expression Like / Not Like 'Pattern to match' Escape 'char';

NOTE:
1 Escape character must be defined
2 Escape char must be used before the special character ('% or _') which has to treated as a Normal Character.
3 The recommended character for Escape is : [!, /, \, $].

Insta

IID	IName
101	Ram_Pavan
150	Dhoni_Hitter
420	Kohli_no_run
302	Mayank_MP2
143	Dravid_Evil

* WAQD IName from Insta table if name Starts with %
  Select IName
  From Insta
  Where IName Like '!%%' Escape '!';

4. WAQTD Ename if the name ends with '_'
```
Select Ename
From Insta
Where Ename like '%!_' Escape '!';
```

5. WAQTD Ename if the name starts with % and ends with %
```
Select Ename
From Insta
Where Ename like '!%%!%' Escape '!';
```

6. WAQTD Ename if the name has underscore '_'
```
Select Ename
From Insta
Where Ename like '%!_%' Escape '!';
```

7. WAQTD No. of Emps getting Salary Less than Zero In Deptno 10
```
Select Count(*)
From emp
Where Sal < Zero and Deptno=10;
```

2. WAQTD Total Sal needed to pay emp Working as Clerk
```
Select Sum(Sal)
From emp
Where Job='Clerk';
```

3. WAQTD Avg Sal needed to pay all emps
```
Select Avg(Sal)
From emp;
```

4. WAQTD No. of Emps Having 'A' as their first character
```
Select Count(*)
From emp
Where Ename like 'A%';
```

5. WAQTD No. of Emps Working as Clerk or Manager
```
Select Count(*)
From emp
Where Job In ('Clerk','Manager');
```

6. WAQTD Total Sal needed to pay Employees Hired in Feb.
```
Select Sum(Sal)
From emp
Where Hiredate like '%Feb%';
```

7. WAQTD No. of Emps Reporting to 7839 (MGR)
```
Select Count(*)
From emp
Where Mgr=7839;
```

8. WAQTD no. of emps Getting Comm in Deptno 30
```
Select Count(*)
From emp
Where Comm is Not Null And Deptno=30;
```

9. WAQTD Avg Sal, Total Sal, No. of Emps and maximum Sal given to Emps Working as President
```
Select Avg(Sal), Sum(Sal), Count(*), Max(Sal)
From emp
Where Job='President';
```

10 WAQTA no of emps having 'A' in their names

```
Select Count(*)
from emp
Where ename like '%A%';
```

Assignment

11 WAQTA Number of Emps And Total Salary needed to pay the emp who have 2 consicutive L's in their name

->

```
Select Count(*), Sum(Sal)
from emp
Where ename like '%LL%';
```

12 WAQTA Number of Distinct Department present in emp Table

```
Select Count(Distinct Dept)
from emp;
```

13 WAQTA Number of Emps having character '_' in their name

```
Select Count(*)
from emp
Where Ename like '%_%'
```

14 WAQTA Number of emps Having atleast 2 percentage in their name

15 WAQTA TOTAL Salary Given to the Emps Working As Clerk In Dept 30

```
Select Sum(Sal)
from emp
Where Job='Clerk' AND Deptno=30;
```

16 WAQTA Max Salary Given to Emps Working As Analyst

```
Select Max(Sal)
from emp
Where Job='Analyst';
```

17 WAQTA Number of Distinct Salaries Present in emp Table

```
Select Count(Distinct Sal)
from emp;
```

18 WAQTA Number of Distinct Jobs present in emp Table

```
Select Count(Distinct Job)
from emp;
```

19 WAQTA AVG SALARY Given To The Clerk

```
Select Avg(Sal)
from emp
Where Job='Clerk';
```

20 WAQTA MIN Sal Given to the Emps Who Work in Dept 10 As Manager or A Clerk

```
Select Min(Sal)
from emp
Where Job In('Manager','Clerk') AND Deptno=10;
```

Group-by clause:- It is used to group the records

Syntax:-
SELECT group-by-Expression / group-by-function
FROM table-name
[WHERE < filter-condition >]
GROUP By Column-name / expression;

Order of Execution
1- FROM
2- WHERE (If used) [ROW by ROW]
3- GROUP-BY [ROW by ROW]
4- SELECT [GROUP-BY-GROUP]

NOTE:-
- Group clause is used to group the records.
- It Executes Row by Row
- In group by clause We can use Column-name / Expression or an argument
- After the Execution of group by clause We are getting groups
- Therefore any clause Executes after the group by it will Executes group by group
- Without Where clause also We can use group by clause

* WAQTO No of Emps working in each department

Group function
① Select Count(*), Deptno → Group by Expression
② From emp
② Group by Deptno;
Dept column-name

output of from clause

Emp Empno	Ename	Sal	Deptno
1	ALLEN	1900	20
2	JAMES	950	30
3	MILLER	1950	20
4	BLAKE	2900	10
5	JONES	2700	30
6	FORD	1800	10
7	KING	3000	20

output of Group By clause

1	ALLEN	1900	20
3	MILLER	1950	20
7	KING	3000	20
2	JAMES	950	30
5	JONES	2700	30
4	BLAKE	2900	10
6	FORD	1800	10

Count(*)

output of Select clause

Count(*)
3
3
2

output of Group by clause

* WAQTD Max Sal given to the employees in each Job
```
Select max(Sal), Job
From emp
Group by Job;
```

* WAQTD Max & Min Salary given to emps working as a manager or Salesman in each department
```
Select Max(Sal), Min(Sal), deptno
From emp
Where Job In('manager', 'Salesman')
Group by deptno;
```

* WAQTD No of Emps Working in each department except 30 Employees from Deptno 30
```
Select Count(*), Deptno
From emp
Where Deptno != 30
Group by Deptno;
```

* WAQTD Avg Sal given to emp in a Job wise
```
Select Avg(Sal), Job
From Emp
Where Group by Job;
```

1. WAQTD No of Emps working in each department except president
```
Select Count(*)
From emp
Where Job != 'President'
Group by deptno;
```

2. WAQTD Total Sal Needed to pay All Emps in Each Job
```
Select Sum(*), Job
From emp
Group by Job;
```

3. WAQTD No of Emps Working as manager in each dept
```
Select Count(*)
From emp
Where Job='Manager'
Group by Deptno;
```

4. WAQTD Avg Sal needed to pay all empt in Each Dept Excluding the emps of Deptno 20
```
Select Avg(Sal)
From emp
Where Deptno!=20
Group by Deptno;
```

5. WAQTD No of Emps having Char 'A' in their Names in Each Job
```
Select Count(*), Job
From emp
Where Ename Like '%A%'
Group by Job;
```

6. WAQTD No of Emps and Avg Sal needed to pay the Emps who Sal > 2000 in Each dept
```
Select Count(*), Avg(Sal)
From Emp
Where Sal > 2000
Group by deptno;
```

7 WAQTA Total Sal needed to pay and No. of Salesman in each Dept

```
Select sum(sal),count(*)
from emp
Where Job='Salesman'
Group by deptno;
```

8 WAQTA No. of Emps with their max Sal in each Job

```
Select Max(sal),count(*)
from emp
Group by Job;
```

9 WAQTA Max Sal given to an Emp Working in each dept

```
Select Max(sal)
from emp
Group by deptno;
```

10 WAQTA No. of Times the Salary Have Been Repeated in Emp table

```
Select Count(*)-1,Sal
from emp
Group by Sal;
```

4 Having Clause:- It is used to filter the groups

Syntax:

```
SELECT group-by-expression / group_function
From Table-name
[Where <filter_Condition>]
Group By <col-name/expression
Having <group-filter-condition>;
```

"ORDER of Execution"

1- FROM
2- WHERE (Row-By-Row)
3- GROUP BY (Row-By-Row)
4- HAVING (or By 3)
5- SELECT (or By 3)

Emp

Empno	Ename	Sal	Deptno
1	ALLEN	1500	30
2	BRAKES	1950	10
3	JAMES	5900	10
4	MILLER	1900	20
5	FORD	900	30
6	ADAMS	2500	20
7	TURNER	25	10
8	KING	1900	10

output of From clause

Empno	Ename	Sal	Deptno	T/F
1	ALLEN	1500	30	T
2	BRAKES	1950	10	T
3	JAMES	5900	10	T
4	MILLER	1900	20	T
5	FORD	900	30	T
6	ADAMS	2500	20	T
7	TURNER	25	10	F
8	KING	1900	10	F

UNDER EXECUTION

output of Where clause

Empno	Ename	Sal	Deptno
2	BLAKE	1950	10
3	JAMES	5900	10
4	MILLER	1900	20
5	FORD	1900	30
6	ADAMS	2500	20

output of Group By clause.

10

Empno	Ename	Sal	Deptno
2	BLAKE	1950	10
3	JAMES	5900	10

20

Empno	Ename	Sal	Deptno
4	MILLER	1900	20
6	ADAMS	2500	20

30

Empno	Ename	Sal	Deptno
5	FORD	1900	30

Group filter Condition
Count(*)>2

Output of Having clause

* CLARK 1450 10
* JAMES 3800 30

30

4 MILLER 1900 30
6 ADAMS 2500 30

Output of Select clause

Count(*)	Deptno
6	10
2	20

NOTE:-
- Having clause is used to filter the groups
- Having clause execute group by group
- In Having clause we can use group filter condition as an argument (Multi Row Funct.)
- Without group by clause we cannot use Having clause

* WAQTD No. of Empt earning Sal greater than 1500 atleast Two empt are working in Each Department.

```
Select Count(*), Deptno  -- Group by Expression
From emp
Where Sal > 1500         -- Filter Condition
Group by Deptno
Having Count(*) >= 2;
       Group filter condition
       MRF()
```

* WAQTD Max Sal given to empt within are designing as Clerk in a Salesman atleast 3 empt are working in Each department.

```
Select max(Sal), Deptno
From emp
Where Job In ('clerk', 'Salesman')
Group by Deptno
Having Count(*) >= 3;
```

* WAQTD Total Sal and no of empts in each Job exactly four empt are working init.

```
Select Sum(Sal), Job Count(*), Job
From emp
Group by Job
Having Count(*) = 4;
```

* WAQTD Max & Min Sal given to empt in each department where Max Sal is less than 1000 & max Sal > 2000

```
Select max(Sal), min(Sal)
From emp
Where Sal
Group by Deptno
Having MIN(Sal) < 1000 AND MAX(Sal) > 2000;
```

* WAQTD Sal which are repeted in emp table.

```
Select Count(Sal) Sal
From emp
Group by Sal
Having Count(*) > 1;
```

1. WAQTO No of Emps Hired during year of 81 Atleast 3 empt are working in each job
Select Count(*)
From emp
Where hiredate like '%81'
Group by Job
Having Count(*) >= 3;

Assignment

1. WAQTO DNO and Number of Emp working in Each Dept If there Are Atleast 2 Clerks in Each dept
Select Count(*), Deptno
From emp
Where Job = 'Clerk'
Group by Deptno
Having Count(*) >= 2;

2. WAQTO DNO And Total Salary needed to pay all Emp In Each dept If there are atleast 4 Emps in Each dept
Select Sum(Sal), deptno
From emp
Group by deptno
having Count(*) >= 4;

3. WAQTO No of Emp Earning Sal more than 1200 in Each Job And the total Sal needed to pay Emp of Each Job must Exceed 3800.
Select Count(*), Sum(Sal)
From emp
Where Sal > 1200
group by Job
having Count(*) > 3800;

4. WAQTO Deptno and no of Emp Working only If there are 2 Emp Working in Each Dept AS Manager
Select Count(*), Deptno
From tablename emp
Where Job = 'Manager'
Group by Job
Having Count(*) >= 2;

5. WAQTO Job And max Sal of emp in Each Job If the max Sal Exceeds 2600
Select Max(Sal), Job
From emp
Group by Job
Having max(Sal) > 2600;

6. WAQTO the Salaries which are Repeated in Emp table
Select Sal
From emp
Group by Sal
Having Count(*) > 1;

7. WAQTO The Hiredate which are Duplicated in Emp table:
Select Hiredate
From emp
Group by Hiredate
Having Count(*) > 1;

* WAQTO AVG SALARY of each dept if Avg Sal is less than 3000

```
Select Avg(sal)
from emp
Group by deptno
Having Avg(sal) <3000;
```

9. WAQTO Deptno if there are atleast 3 emp in each Dept whos name has char 'A' or 'S'

```
Select deptno
from emp
where ename like '%A%' or '%S%'
Group by deptno
Having Count(*) >= 3;
```

10 WAQTO MIN AND MAX SAL of each Job if min Sal is more than 1000 and max Sal less than 5000

```
Select min(sal), max(sal)
from emp
group by Job
having min(sal)>1000 and max(sal)<5000;
```

* ORDER BY

- Order by clause is used to arrange the records either in Ascending Order or in descending order
- By Default Order By clause sorts the records in Asc order
- Order by clause executes at the last
- Order by clause executes after Select clause
- Order by clause executes Row by Row

Syntax:-

```
Select group-by-expression [group-function
from tablename
[Where <filter condition>]
[Group By Col-name/expression]
[Having <group-filter-Cond/Exp>]
#ORDER By Colomname/expression [ASC]/[DESC];
```

ORDER of EXECUTION

1- FROM [
2- WHERE [Row by Row]
3- GROUP BY [Row by Row]
4- HAVING [Te by e]
5- SELECT [e by e]
6- ORDER BY [Row by Row]

* WAQTO Sal of an Emp arrange the Emp Sal in the ascending order

```
Select Sal
from emp
order by Sal asc;
```

* WAQTO Name of Emp who are working as a manager or Salesman arrang Emp name in descending order

```
Select Ename
from emp
where Job In('manager' ~~and~~, 'salesman')
order by Ename desc;
```

* WAQTO No. of Emps in Job if Emp name has chars and working in Each Job arrange the Job column in a descending order

```
Select Count(*), Job
from emp
where Ename like '%A%'
Group By Job
order By Job DESC;
```

* WAQTD Max Sal given to Emps and Deptno allocate to emps are working in each department Arrange the values in dep wise

```
Select Max(Sal), Deptno
from emp
Group By Deptno
Having Count(*) >=2
Order By Deptno;
```

* Subquerys: A query return inside another query is known as Subquery.

Working Flow:

Result outputs (outer Query) ⟷ Dependent ⟷ (INNER QUERY)

- Let us Consider two query outer query & Inner query
- In Subquery inner query will execute first and generates a output.
- The output of an inner query is sending as a input to the outer query
- With the help of this input outer query will execute and generates an result and an output table
- Therefore outer query is dependent on inner query

* When and why We use Subquerys?

Case1

When there is unknown condition present in a query, We use Subquerys

Emp:

Ename	Sal	Deptno
ADAMS	1900	20
JONES	1950	10
BLAKE	9850	30
FORD	4000	10
KING	1280	30

Case1: Unknown Condition

* WAQTD Names of Emps who are earning Sal greather than 1500

```
Select Ename
From Emp
Where Sal > 1500;
```

Direct Condition
or
Known Condition

* WAQTD Names of an emps who are earning Sal greather than Jones.

```
Select Ename
from Emp
Where Sal > 'JONES';
```

Indirect Condition
or
Unknown Condition

```
Select Ename
from emp
Where Sal > (Select Sal
from emp
Where Ename = 'JONES');
```

->	1900	F
->	1950	F
->	9850	T
->	4000	T
->	1280	F

JONES	1950	10

* WAQTD Name & Hiredate of an emp if Emps are hired before the Miller

```
Select ename,hiredate
from emp
where Hiredate < (Select Hiredate
                  from emp
                  where Ename='Miller');
```

* WAQTD Names of employees who are working in the same designation of Adams.

```
Select ename
from emp
where Job= (select Job
            from emp
            where Ename='adams');
```

* WAQTD Details of emps who are working in the same dept of Jones

```
Select *
from emp
where deptno (Select deptno
              from emp
              where Ename='JONES');
```

* WAQTD Details of Emps who are working as a manager and hired after the Smith

```
Select *
from emp
where Job='Manager' AND hiredate>
      (Select hiredate
       from emp
       where Ename='Smith');
```

* WAQTD Name & Sal of Emps if emps are earning Sal less than the Blake and emp name has Char A

```
Select Ename,Sal
from Emp
where Ename Like '%A%' AND Sal < (Select Sal
                                  from emp
                                  where Ename='BLAKE');
```

* WAQTD Details of emps who are earning sal greater than Jones and less than King

```
Select *
from emp
where Sal > (Select Sal      AND Sal < (Select Sal
             from emp                   from emp
             where Ename='Jones')       where Ename='king');
```

* WAQTD Details of Emp who are hired after Allen Adams & before a Emp hired year of 81.

```
Select *
from emp
where Hiredate Like '%81' AND Hiredate > (Select hiredate
                                          from emp
                                          where Ename='ALLEN') AND Hiredate < (Select hiredate
                                                                               from emp
                                                                               where Ename='ADAMS');
```

* WAQTD-Details of emp along with Annual Sal if Emps are working in same dept of Turner and same designation of Martin and earning Sal greater than Smith

```
Select emp.*, Sal*12
from emp
where deptno = (Select deptno
                from Emp
```

Job
!where ename = 'Turner' AND Job = (Select ename
from emp
!where Ename = 'MANAGER');

Sal
Job > (Select Sal
from emp
!where Ename = 'SMITH');

CHEDULE

WEDNESDAY	THURSDAY	FRIDAY

my self

Name: Aman Raj

Class:

School / College:

Email:

Index

Date	Page No.	Subject:	Sign	Remark